Kitchen Story an E-Commerce Application

This document contains sections for:

- The code for this project is hosted at **https://github.com/Dinesh123527/simpli-phase_4_project_kitchen_story**
- This project is developed by **G V Narasimha Raju.**

Sprints planning and Task completion:

The project is planned to be completed in 2 sprints. Tasks assumed to be completed in the sprint are:

1st Sprint:

- Creating the flow of the application.
- Initializing the git repository to track changes as development progresses.
- Creating a React JS application to fulfill user requirements.
- Adding the Required Packages used for the application.
- Creating React Elements and Configuring Tailwind CSS.

2nd Sprint:

- Implemented the Redux and Reducer for State Management.
- Creating Form and Validations to the Form.
- Creating new Firebase Project and Initiating it.
- Configuring the Firebase Project and creating new modules in firebase store for authentication, database for storing the food items.

- Implement the Logic for Admin Login using Google Authentication through Firebase Store and creating new Items in Firebase Store with Documents.
- Getting all the Food Items from Fire Base Store and also Implementing Filtering the Menu Items based on Menu Type.

Core concepts used in the project:

• React JSX: JSX provides you to write HTML/XML-like structures (e.g., DOM-like tree structures) in the same file where you write JavaScript code, then preprocessor will transform these expressions into actual JavaScript code.

• React Components: A Component is considered as the core building blocks of a React application. It makes the task of building UIs much easier.

• React State: The state is a built-in React object that is used to contain data or information about the component. A component's state can change over time.

• React Props: In ReactJS, the props are a type of object where the value of attributes of a tag is stored. The word "props" implies "properties", and its working functionality is quite similar to HTML attributes.

• React Events: There are many events that get triggered by the system such as clicking, page loading, keypress, scrolling, etc. React provides event handling which is similar to event handling on DOM elements. React events are synthetic events, a cross-browser wrapper around the browser's native event. It works the same across all browsers.

• React Routing: React Router is a JavaScript framework that lets us handle client and server-side routing in React applications. It enables the creation of single-page web or mobile apps that allow navigating without refreshing the page. It also allows us to use browser history features while preserving the right application view.

• React Hooks: Hooks are functions that let you “hook into” React state and lifecycle features from function components. Hooks don't work inside classes they let you use React without classes.

UseEffect(), UseState() etc.

• Reducer: Reducers, as the name suggests, take in two things: previous state and an action. Then they reduce it (read it return) to one entity: the new updated instance of state. So reducers are basically pure JS functions which take in the previous state and an action and return the newly updated state.

• API: API stands for application programming interface, which is a set of definitions and protocols for building and integrating application software.

Technologies and Tools Used in the Project

- **React JS:** To create Kitchen Story an E-Commerce Application

- **Tailwind CSS:** Tailwind CSS is basically a utility-first CSS framework for rapidly building custom user interfaces.

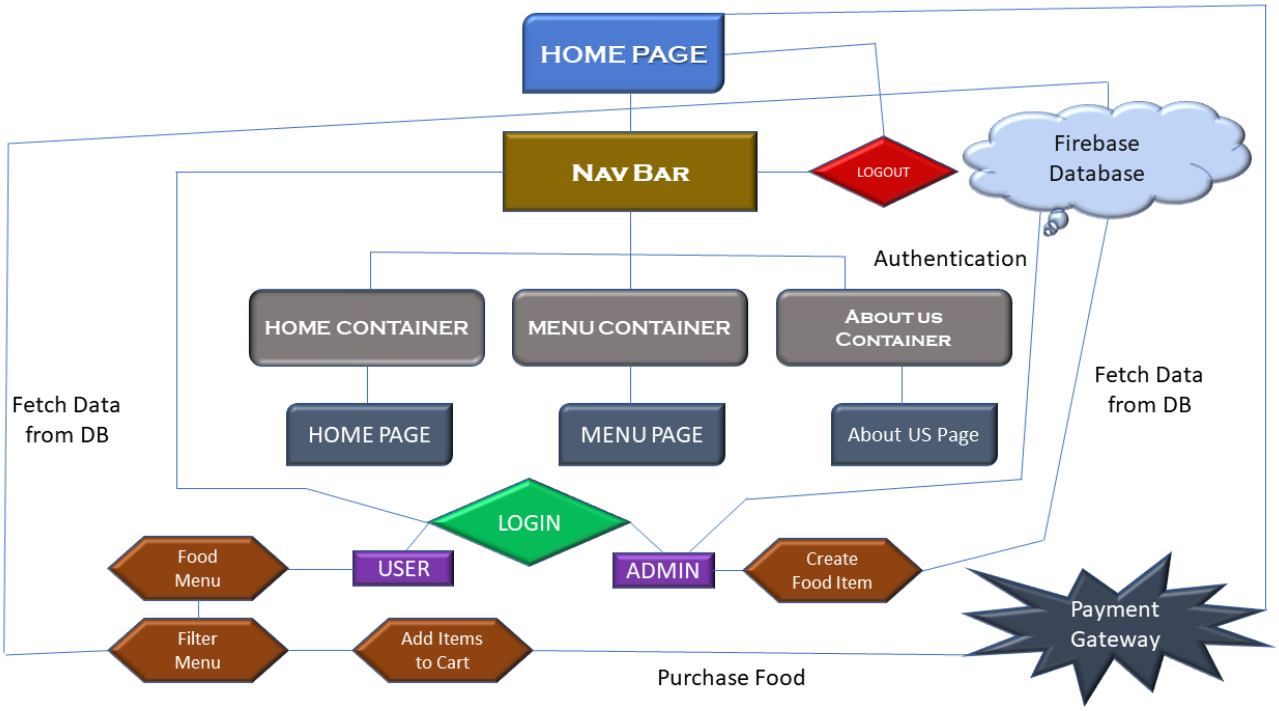
- **Redux:** Redux is a predictable state container for JavaScript apps. As the application grows, it becomes difficult to keep it organized and maintain data flow. Redux solves this problem by managing application’s state with a single global object called Store.

- **Firebase**: Firebase is a Backend-as-a-Service (BaaS) app development platform that provides hosted backend services such as a Realtime database, cloud storage, authentication, crash reporting, machine learning, remote configuration, and hosting for your static files.

- **Visual Studio Code:** Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and

version control. It aims to provide just the tools a developer needs for a quick code-build-debug cycle.

Flow of the Application:



Demonstrating the product capabilities, appearance, and user interactions:

To demonstrate the product capabilities, below are the sub-sections configured to highlight appearance and user interactions for the project.

- ✓ **Creating new React JS project using npx create-react-app project name.**
- ✓ **Add the required packages to the project via node package manager npm command.**

- ✓ **Open the React Project in that go to package.json to view the added dependencies.**
- ✓ **Now Run the code using npm command and the application will be deployed in the default port of localhost:3000.**
- ✓ **Initialize the Firebase Project through npm firebase install command and configure the firebase config file.**
- ✓ **Now Create new Project in Firebase Store and Init in your React Application through firebase init command.**
- ✓ **Pushing the code to the GitHub repository.**

Unique Selling Points of the Application:

- ✓ This was an E- Commerce Application developed using React JS where there is a user and an admin as well to manage the Application.
- ✓ The Application has both Functionality and a real time Authentication also using Google Authentication using Firebase as provider and the admin can Create and Delete the Food Items, but Where as user can Only View the menu and Filter the Food Items.
- ✓ The Admin User Can Purchase any Item available in the Cart and coming to the cart Section the Cart value is updated based on the Food Item Selected Dynamically he can clear the cart and select new Items and User also can purchase Items through this.

Conclusions:

Further enhancements to the application can be made which may include:

- Separated the Role's for User and Admin but only Admin has Login Part a normal user cannot login to the application and purchase desired Items.
- Need to Validate the User's Login also but the user can only view, filter available Food Items can also buy them.

- Need to Implemented the Authenticated Routes for Different Roles in the Application and Need to add the Real Time Payment Gateway Integration and update the cart section based on the availability of the product.